Comparison Study of Software Testing Methods and Levels- A Review

Deepti Kapila Grover

M.Tech, Dept. of Computer Science, Assistant professor in LCET, Katani Kalan, India

ABSTRACT: Software testing is an activity which is aimed for evaluating an attribute or capability of a program and ensures that it meets the required result. There are many approaches for software testing but the correct method used for testing is necessary to carry out all the functions of software item or program. The levels of software testing are also the important part of software item. To find out errors from software item, the necessary conditions are carried out under testing methods and testing levels for testing technique. In this paper, various methods and levels of software testing are described. This paper includes comparative study of important software testing methods and software testing levels.

KEYWORDS: Security testing, methods, functional testing, regression, validation

I. INTRODUCTION

Software testing is a process or the series of processes designed to verify computer code does what it was designed to do. Testing is also viewed as process of analysing software item to detect the difference between existing and required conditions and to evaluate the features of a software item [1, 2]. To find the errors in a product, the technique testing is used in which errors are find out to correct execution of the software. In other words, Software testing is an activity to evaluate the program or system and determining that it meets its desired results [3]. Testing is the process of demonstrating that errors are not present in program or software item. The program performs its functions correctly after testing the program or product. Test condition describes an input description and an expected output description. The inputs are of two types named as pre-conditions and actual inputs. The description of these two is given below:

a. **Pre-conditions:** Circumstances that hold prior to test case execution.
b. **Actual inputs:** that is identified by some testing methods.

During testing, we set the necessary preconditions, given required inputs to a program and compare the observed output to know the outcome of test case**.** If observed and expected outputs are different, it is considered as failure but if both are same there is no failure.
Software testing consists of the dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite execution domains, against the expected specified behavior. The term dynamic here means that testing implies executing the program on values (inputs). The figure1 shows two types of inputs used in test conditions of software item or program.
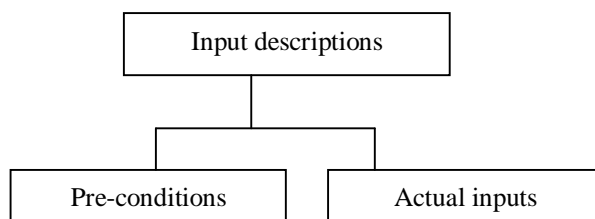

Figure 1: Description of inputs in test case of testing

II. OBJECTIVES OF SOFTWARE TESTING

Basically software testing is the way to detect the errors from software item of program where conditions are tested to make the product more efficient and cost constructive. The main purpose of software testing is quality assurance, reliability, validation and verification. To maintain the originality and to work the functions of software item properly, the testing is required. So as it is needed, there is couple of objectives for testing which are mentioned below:

- A successful testing is considered if it uncovers discovered errors.
- Better the software works, the more efficiently it can be used.
- It includes correctness and completeness of software.
- It ensures the quality of software item by testing software item under some conditions.
.

III. TERMS ASSOCIATE WITH TESTING

The main purpose of software testing is quality assurance, reliability, validation and verification. These terms are used in software testing and important as well. Description of terms is given below:

Quality Assurance: All those planned or systematic actions necessary to provide confidence that a product or service is of the type and quality needed and expected by the customer.

Reliability: It is another term of consistency. If one person takes same personality test several times and always gets the same result is called reliable state.

Verification: It is the process of evaluating a system to determine whether the given product satisfied the conditions imposed at the start of that phase.

*Validation***:** It is the process of evaluating a system during or at the end of the development process to determine whether it satisfies the conditions.

The comparison of terms verification and validation is described in Table 1 given below:

Verification	**Validation**
It ensures that you built the product right	It ensures that you built the right product
It ensures that product has been built according to user requirements and design specifications	It ensures that the product actually meets the user needs and that the specification were correct in the first place
Satisfies the conditions imposed at the start of the phase	Satisfies the conditions during or at the end of the development process

Table 1: Comparison study of terms verification and validation

Above table differentiate the verification and validation terms in more understandable form. This comparison study of verification and validation defines how these two terms are similar and differ from each other. Moreover, quality assurance and reliability are two factors for software testing methods and levels but the factors verification and validation are commonly used when software testing part is done.

IV. **SOFTWARE TESTING- METHODS**

This part describes that there are couple of methods used for software testing. In this, the description of every testing is given one by one. The following parts are types and descriptions of software testing.

1. Black Box Testing**:** This type of testing includes only interior working of the software. In this, the tester does not access the source code. The tester will interact with the system user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. The representation of black box testing is given in figure 2 below:

Advantages:
   ➢ Well suited and efficient for large code segments
   ➢ Clearly separate user perspective from the developer perspective through visibly defined roles
   ➢ Without knowledge of implementation and programming, large number of skilled testers can test

Disadvantages:
   ➢ Limited coverage
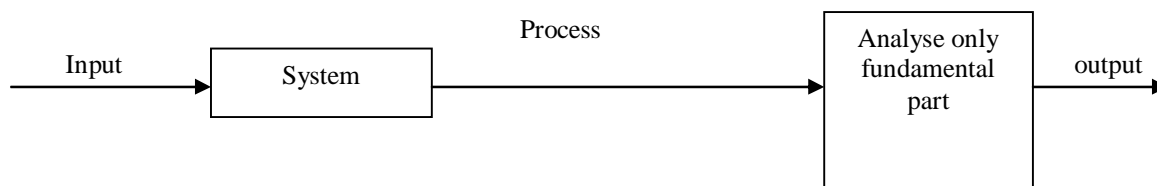   ➢ Inefficiently testing due to limited knowledge

Figure 2: Representation of black box testing

2. **White Box Testing:** It is also called open testing or glass testing. In this testing, a tester needs to know the internal working code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving properly. It is detailed investigation of internal logic and the structure of the code. The representation of white box testing is shown in figure 3 given below:

Advantages:
   ➢ Becomes easy to find out errors with the knowledge of source code
   ➢ Helps in optimizing the code
   ➢ Efficient application as extra lines can be removed from code

Disadvantages:
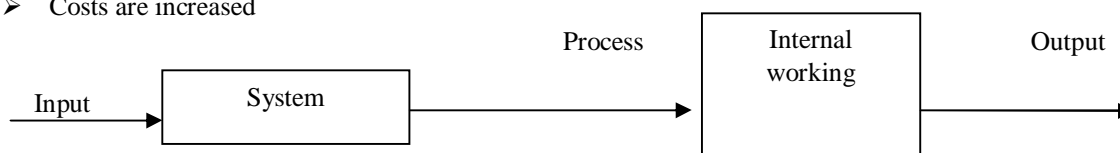   ➢ Hidden errors may create problems sometimes
   ➢ Costs are increased

Figure 3: Representation of white box testing

3. **Grey Box testing:** It is a technique includes limited knowledge if internal working. Unlike black box testing, where the tester only tests the application interface, in grey box testing, the tester has access to design documents and database. Having the knowledge the tester can prepare better test data and test scenarios while making a test plan.

Advantages:
- ➢ Combined benefits of black box and white box testing
- ➢ Based on limited knowledge the tester can design excellent test plan
- ➢ Does not rely on source code, as based on interface definition and functional specifications

Disadvantages:
- ➢ Unrealistic
- ➢ Tests can be redundant

The above parts of software testing describe overall structure of methods of testing. To understand in a more convenient manner, the comparison table of white, black and grey box testing is given below as table 2.

Black Box Testing	White Box Testing	Grey Box testing
No need of internal code	Full knowledge of internal code	Limited knowledge of internal code
Known as closed box, data driven or functional testing	Known as clear box, structural or code based testing	Known as translucent testing
Exhaustive and time consuming	Exhaustive and time consuming	Partly time consuming
External expectations	Based on high level database and DFD's	Internal workings
Not suited for algorithm testing	Suited for algorithm testing	Not suited for algorithm testing

Table 2: Comparison of black box, white box and grey box testing

V. SOFTWARE TESTING LEVELS

There are different levels for software testing. In this part, we will conclude all levels of software testing with description. Levels of testing can include different methodologies. The main levels of software testing are:

Functional testing: type of black box testing on the specifications of the software that is to be tested.

Non functional testing: is based on testing an application form its non functional attributes.
There are levels of functional testing which are described below one by one.

Levels of functional testing

a) *Unit testing:* This testing is performed by respective developers on the individual units of source code assigned areas. The goal of unit testing is to isolate each part f the program and show that individual parts are correct according to requirements and functionality.

b) *Integration testing:* It is combined part of an application to test whether all functions are correct or not. This technique is based on: Top down integration and bottom up integration. Top down means the testing begins with unit testing. Bottom up means highest modules are tested first.

c) *System testing:* The purpose of system testing is to validate an application's accuracy and completeness in performing the function as designed.

d) *Regression testing:* It is used to ensure that a change like bug fix should not result in another fault being uncovered in the application. This testing minimizes the gap when an application with changes made has to be tested.

e) *Acceptance testing:* This testing is conducted by quality assurance team. The quality assurance team will have a set of pre-written scenarios and test cases that will be used to test the application.

f) *Alpha testing:* This testing is conducted at the developer's site by a customer. Unit testing, integration testing and system testing when combined together is known as alpha testing.

g) *Beta testing:* This testing is conducted by customers/end users at either sites. Unlike alpha testing, developer is not present here. The comparison of alpha and beta testing is shown in table 3.

Levels of Non-functional testing

a) *Performance testing:* It is used to identify any bottlenecks and performance issues in software. Number of performance issues can be there like speed, stability, scalability etc*.*

b) *Load testing:* This tests the behavior of software by applying maximum load. This type of testing identifies the maximum capacity of software and its behavior at peak time.

c) *Performance Stress testing:* This is a technique to test the behavior of software under abnormal conditions. This testing can be performed by testing scenarios like shut down or restart of network ports randomly, turning the database on/off etc.

d) *Usability testing:* This is used o identify an errors and improvements in software by observing through their usage and operation.

e) *Security testing:* Testing software in order to identify any flaws and gaps from security point of view\.security testing should ensure authentication, integrity, authorization etc.

f) *Portability testing:* This testing ensures reusability and that it can be moved from other software well. This testing can be considered as one of the part of system testing.

Basically, the alpha and beta testing is done for software item or product but it entirely depends on the customer needs and requirements that he/she is comfortable with whatever type of testing levels and methods. Commonly number of errors is finding out through levels like functional levels. Non functional levels are also part, but not so effective levels for software testing.

Alpha testing	Beta testing
Performed by the developers at the software development site	Performed by the customers at their own site
Performed by independent testing team	Not performed by independent testing team
Performs in virtual environment	Performs in real time environment
Not open to market and public	Open to market and public
Performs within the organization	Performs outside the organization
Category of both testing techniques i.e white box and black box testing	Only black box testing

Table 3: Comparison study of alpha and beta testing

IV. CONCLUSION

Software testing is viewed as process of analysing software item to detect the difference between existing and required conditions and to evaluate the features of a software item. There are many approaches to test the software but the adoption of correct and complete method and level is important to carry out all the functions properly in software item or in program. The method of software testing includes black box testing, white box testing and grey box testing. The terms validation and verifications are also included for better understanding of software methods. The comparison study fully satisfies the differentiation between black box, white box and grey box testing. Either there is couple of levels in testing like functional and non-functional; the entire testing technique must be suitable and easy to perform according to user or developer both. Functional and non-functional levels highlight various testing parts to more efficient and better performance of software during testing.

REFERENCES

1. Security testing-wikipedia the free encyclopediaavailable at http://en.wikipedia.org/wiki/securitytetsing
2. White box testing from wikipedia, the freeencyclopedia
3. Software testing for wikipedia available athttp://en.wikipedia.org/wiki/grey_box_testing#grey_box_tetsing
4. Lessons Learned in Software Testing, by C. Kaner, J. Bach, and B. Pettichord
5. Shilpa Testing Computer Software, by C. Kaner, J. Falk, and H. Nguyen
6. Effective Software Testing, by E. Dustin
7. Software engineering, by Roger Pressman